

통계 분석

```
def parse_url(url, css_selector):  
    r = requests.get(url)  
    soup = BeautifulSoup(r.content, 'xml')  
    s = soup.select_one(css_selector)  
    with open('article.txt', 'w+') as f:  
        f.write(s.text.strip())  
    return f.name
```

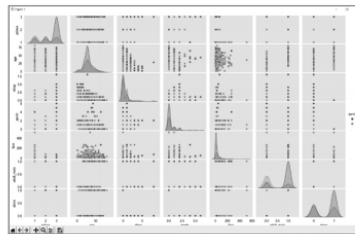
소프트웨어융합대학원
진혜진

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

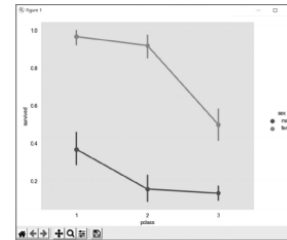
■ 분석 미리보기

| 타이타닉호 생존율 분석하기 | |
|----------------|--|
| 목표 | 타이타닉호 승객 변수를 분석하여 생존율과의 상관관계를 찾는다. |
| 핵심 개념 | 상관 분석, 상관 계수, 피어슨 상관 계수, 히트맵 |
| 데이터 수집 | 타이타닉 데이터: seaborn 내장 데이터셋 |
| 데이터 준비 | 결측치 치환: 중앙값 치환, 최빈값 치환 |
| 데이터 탐색 | 1. 정보 확인: info() 2. 차트를 통한 데이터 탐색: pie(), countplot() |
| 데이터 모델링 | 1. 모든 변수 간 상관 계수 구하기 2. 지정한 두 변수 간 상관계수 구하기 |
| 결과 시각화 | |

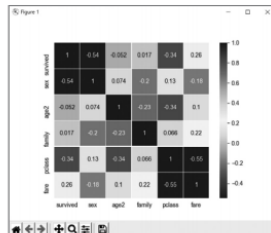
1. 산점도를 이용한 시각화



2. 특정 변수 간 상관관계 시각화



3. 히트맵을 이용한 시각화



02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 분석 미리보기

- 타이타닉호의 생존자와 관련된 변수의 상관관계를 찾아봄
- 생존과 가장 상관도가 높은 변수는 무엇인지 분석
- 상관 분석을 위해 피어슨 상관 계수를 사용
- 변수 간의 상관관계는 시각화하여 분석

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 핵심 개념 이해

■ 상관 분석

- 두 변수가 어떤 선형적 관계에 있는지를 분석하는 방법
- 두 변수는 서로 독립적이거나 상관된 관계일 수 있는데, 두 변수의 관계의 강도를 상관관계 라고함
- 상관 분석에서는 상관관계의 정도를 나타내는 단위로 모상관 계수 ρ 를 사용
- 상관 계수는 두 변수가 연관된 정도를 나타낼 뿐 인과 관계를 설명하지 않으므로 정확한 예측치를 계산할 수는 없음

• 단순 상관 분석

- 두 변수가 어느 정도 강한 관계에 있는지 측정

• 다중 상관 분석

- 세 개 이상의 변수 간 관계의 강도를 측정
- 편상관 분석: 다른 변수와의 관계를 고정하고 두 변수 간 관계의 강도를 나타내는 것

■ 상관 계수 ρ

- 변수 간 관계의 정도(0~1)와 방향(+, -)을 하나의 수치로 요약해주는 지수로 -1에서 +1 사이의 값을 가짐
- 상관 계수가 +이면 양의 상관관계이며 한 변수가 증가하면 다른 변수도 증가
- 상관 계수가 -이면 음의 상관관계이며 한 변수가 증가할 때 다른 변수는 감소
- 0.0 ~ 0.2: 상관관계가 거의 없음
- 0.2 ~ 0.4: 약한 상관관계가 있음
- 0.4 ~ 0.6: 상관관계가 있음
- 0.6 ~ 0.8: 강한 상관관계가 있음
- 0.8 ~ 1.0: 매우 강한 상관관계가 있음

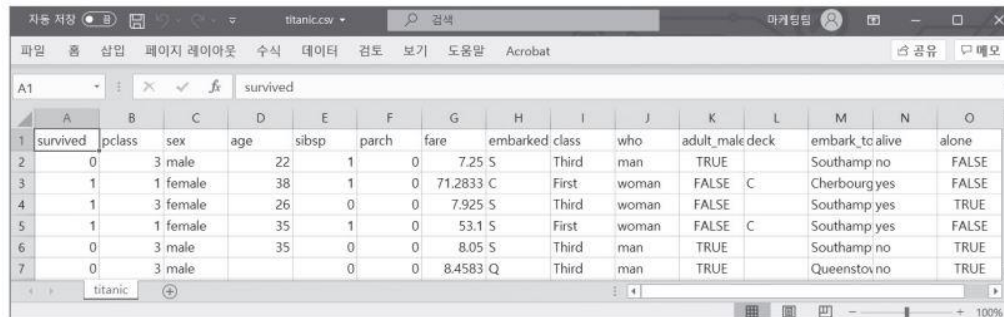
02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 수집

```
01 >>> import seaborn as sns
02 >>> import pandas as pd
03 >>> titanic = sns.load_dataset("titanic")
04 >>> titanic.to_csv('titanic.csv', index = False)
```

- 01행 seaborn 패키지를 로드
- 03행 titanic 데이터를 로드
- 04행 데이터를 CSV 파일로 저장

■ 데이터 준비



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|----------|--------|--------|-----|-------|-------|---------|----------|-------|-------|------------|------|------------|-------|-------|
| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | embark_tci | alive | alone |
| 1 | 0 | 3 | male | 22 | 1 | 0 | 7.25 | S | Third | man | TRUE | | Southamp | no | FALSE |
| 2 | 1 | 1 | female | 38 | 1 | 0 | 71.2833 | C | First | woman | FALSE | C | Cherbourg | yes | FALSE |
| 3 | 1 | 3 | female | 26 | 0 | 0 | 7.925 | S | Third | woman | FALSE | | Southamp | yes | TRUE |
| 4 | 1 | 1 | female | 35 | 1 | 0 | 53.1 | S | First | woman | FALSE | C | Southamp | yes | FALSE |
| 5 | 0 | 3 | male | 35 | 0 | 0 | 8.05 | S | Third | man | TRUE | | Southamp | no | TRUE |
| 6 | 0 | 3 | male | | 0 | 0 | 8.4583 | Q | Third | man | TRUE | | Queenstov | no | TRUE |

그림 7-8 다운로드한 파일(titanic.csv) 열기

- 저장한 titanic.csv 파일을 열어서 데이터 정리 작업이 필요한지 확인

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 준비

```
01 >>> titanic.isnull().sum()
survived      0
pclass        0
sex           0
age          177
sibsp         0
parch         0
fare          0
embarked      2
class         0
who           0
adult_male    0
deck         688
embark_town    2
alive         0
alone         0
dtype: int64

02 >>> titanic['age'] = titanic['age'].fillna(titanic['age'].median())
03 >>> titanic['embarked'].value_counts()
S    644
C    168
Q     77
Name: embarked, dtype: int64

06 >>> titanic['embark_town'] = titanic['embark_town'].fillna('Southampton')
07 >>> titanic['deck'].value_counts()
```

```

C     59
B     47
D     33
E     32
A     15
F     13
G      4
Name: deck, dtype: int64

08 >>> titanic['deck'] = titanic['deck'].fillna('C')
09 >>> titanic.isnull().sum()
survived      0
pclass        0
sex           0
age           0
sibsp         0
parch         0
fare          0
embarked      0
class         0
who           0
adult_male    0
deck          0
embark_town    0
alive         0
alone         0
dtype: int64
```

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 탐색

1. 데이터의 기본 정보 탐색하기

```
01 >>> titanic.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  -
0   survived    891 non-null    int64
1   pclass      891 non-null    int64
2   sex         891 non-null    object
3   age         891 non-null    float64
4   sibsp       891 non-null    int64
5   parch       891 non-null    int64
6   fare 891    non-null      float64
7   embarked    891 non-null    object
8   class 891    non-null      category
9   who 891     non-null      object
10  adult_male   891 non-null    bool
11  deck 891     non-null      category
12  embark_town  891 non-null    object
13  alive        891 non-null    object
14  alone 891    non-null      bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
02 >>> titanic.survived.value_counts()
0    549
1    342
Name: survived, dtype: int64
```

- 01행 타이타닉 데이터의 기본 정보를 확인
- 02행 survived 속성값의 빈도를 확인
- 전체 샘플의 수: 891개이고 속성은 15개
- 샘플 891명 중에서 생존자는 342명이고 사망자는 549명
- pclass, class: 객실 등급
- sibsp: 함께 탑승한 형제자매와 배우자 수
- parch: 함께 탑승한 부모/자식 수
- embarked, embark_town: 탑승 항구
- adult_male: 성인 남자 여부
- alone: 동행 여부를 True/False로 나타냄

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 탐색

2. 차트를 그려 데이터를 시각적으로 탐색하기

```
01 >>> import matplotlib.pyplot as plt
02 >>> f,ax = plt.subplots(1, 2, figsize = (10, 5))
03 >>> titanic['survived'][titanic['sex'] == 'male'].value_counts().plot.
    pie(explode = [0,0.1], autopct = '%1.1f%%', ax = ax[0], shadow = True)
    <matplotlib.axes_subplots.AxesSubplot object at 0x000001DD48E0C648>
04 >>> titanic['survived'][titanic['sex'] == 'female'].value_counts().plot.
    pie(explode = [0,0.1], autopct = '%1.1f%%', ax = ax[1], shadow = True)
    <matplotlib.axes_subplots.AxesSubplot object at 0x000001DD491C35C8>
05 >>> ax[0].set_title('Survived (Male)')
    Text(0.5, 1.0, 'Survived (Male)')
06 >>> ax[1].set_title('Survived (Female)')
    Text(0.5, 1.0, 'Survived (Female)')
07 >>> plt.show()
```

- 01행 차트를 그리기 위해 matplotlib.pyplot를 로드
- [02~07행] 남자 승객과 여자 승객의 생존율을 pie 차트로 그리기
 - 02행 한 줄에 두 개의 차트를 그리도록 하고 크기를 설정
 - 03행 첫 번째 pie 차트는 남자 승객의 생존율을 나타내도록 설정
 - 04행 두 번째 pie 차트는 여자 승객의 생존율을 나타내도록 설정
 - 05행 첫 번째 차트의 제목을 설정
 - 06행 두 번째 차트의 제목을 설정
 - 07행 구성한 차트를 나타낸다.

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 탐색

2. 차트를 그려 데이터를 시각적으로 탐색하기

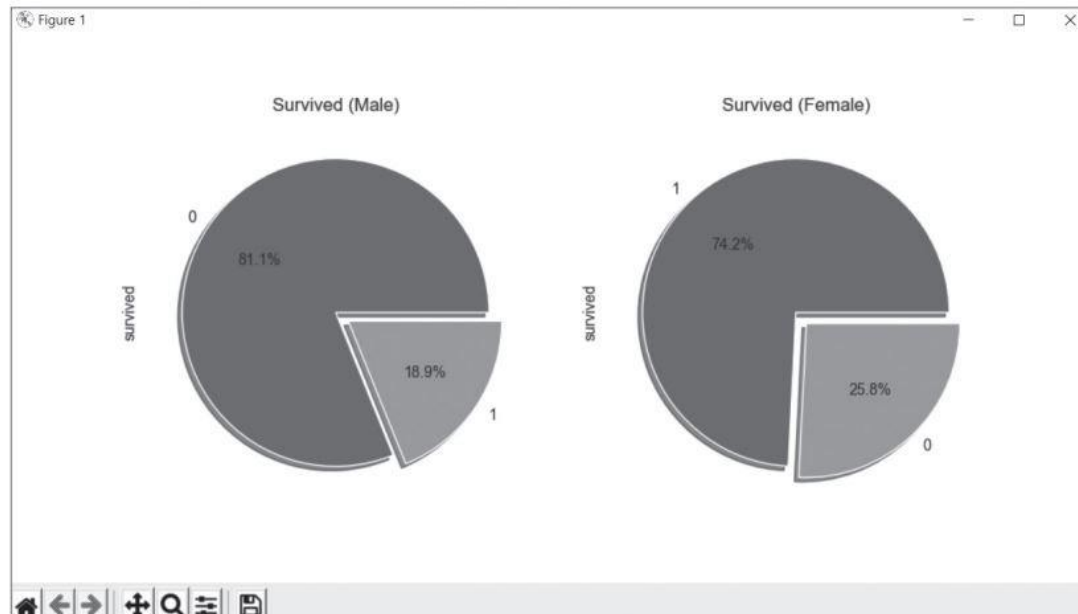


그림 7-9 성별에 따른 생존율 차트

- 남자 승객의 생존율: 18.9%
- 여자 승객의 생존율 74.2%

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 탐색

3. 등급별 생존자 수를 차트로 나타내기

```
01 >>> sns.countplot('pclass', hue = 'survived', data = titanic)
    <matplotlib.axes._subplots.AxesSubplot object at 0x000001DD48E03EC8>
02 >>> plt.title('Pclass vs Survived')
    Text(0.5, 1.0, 'Pclass vs Survived')
03 >>> plt.show()
```

- 01행 pclass 유형 1,2,3을 x축으로 하고 survived =0과 survived =1의 개수를 계산하여 y축으로 하는 countplot을 설정
- 02행 차트 제목을 설정
- 03행 구성한 차트를 나타냄
- 생존자(1)는 1등급에서 가장 많음
- 사망자(0)는 3등급에서 월등히 많음

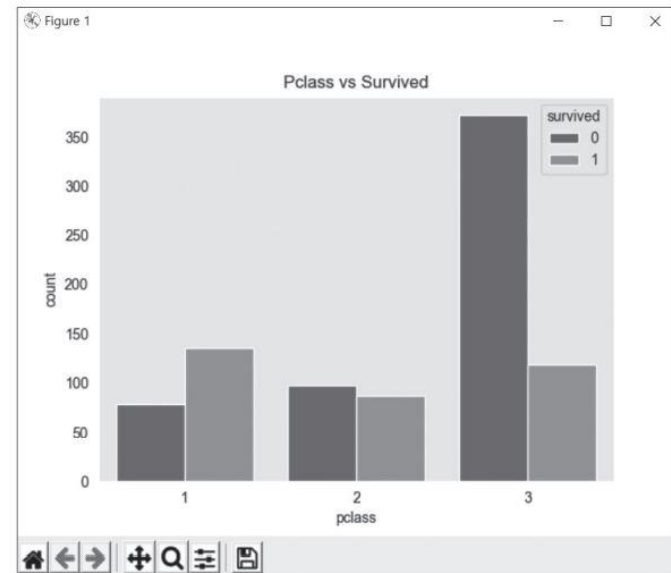


그림 7-10 객실 등급에 따른 생존자 수

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 모델링

1. 상관 분석을 위한 상관 계수 구하고 저장하기

```
01 >>> titanic_corr = titanic.corr(method = 'pearson')
02 >>> titanic_corr
survived    pclass      age      ...    fare  adult_male    alone
survived    1.000000 -0.338481 -0.064910 ... 0.257307 -0.557080 -0.203367
pclass     -0.338481  1.000000 -0.339898 ... -0.549500  0.094035  0.135207
age        -0.064910 -0.339898  1.000000 ...  0.096688  0.247704  0.171647
sibsp      -0.035322  0.083081 -0.233296 ...  0.159651 -0.253586 -0.584471
parch       0.081629  0.018443 -0.172482 ...  0.216225 -0.349943 -0.583398
fare        0.257307 -0.549500  0.096688 ...  1.000000 -0.182024 -0.271832
adult_male -0.557080  0.094035  0.247704 ... -0.182024  1.000000  0.404744
alone      -0.203367  0.135207  0.171647 ... -0.271832  0.404744  1.000000
[8 rows x 8 columns]
03 >>> titanic_corr.to_csv('C:/Users/kmj/My_Python/7장_data/titanic_corr.csv',
index = False)
```

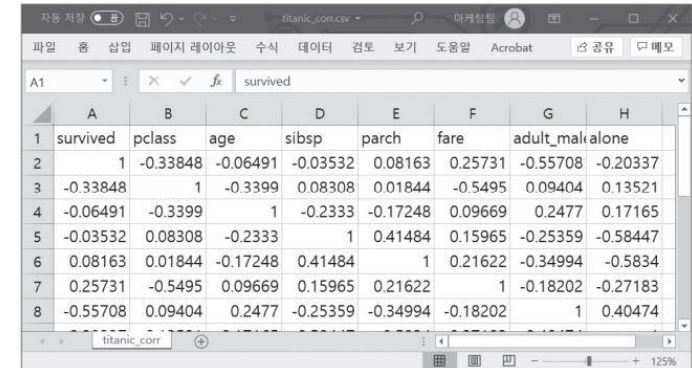
- 01행 피어슨 상관 계수를 적용하여 상관 계수를 구함
- 02행 상관 계수를 출력
- 03행 상관 계수를 CSV 파일로 저장

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 데이터 모델링

2. 상관 계수 확인하기

- 남자 성인(adult_male): 생존(survived)과 음의 상관관계
- 객실 등급(pclass): 음의 상관
- 관계, 객실 요금fare은 양의 상관관계
- 동행 없이 혼자 탑승한 경우(alone): 생존율이 떨어진다는 상관관계



| | A | B | C | D | E | F | G | H |
|---|----------|--------|----------|----------|----------|----------|------------|----------|
| 1 | survived | pclass | age | sibsp | parch | fare | adult_male | alone |
| 2 | | 1 | -0.33848 | -0.06491 | -0.03532 | 0.08163 | 0.25731 | -0.55708 |
| 3 | | | 1 | -0.3399 | 0.08308 | 0.01844 | -0.5495 | 0.09404 |
| 4 | | | | 1 | -0.2333 | -0.17248 | 0.09669 | 0.2477 |
| 5 | | | | | 1 | 0.41484 | 0.15965 | -0.25359 |
| 6 | | | | | | 1 | 0.21622 | -0.34994 |
| 7 | | | | | | | 1 | -0.18202 |
| 8 | | | | | | | | 1 |

그림 7-11 titanic_corr.csv 파일에서 상관 계수 확인

3. 특정 변수 사이의 상관 계수 구하기

```
01 >>> titanic['survived'].corr(titanic['adult_male'])  
-0.5570800422053259  
02 >>> titanic['survived'].corr(titanic['fare'])  
0.2573065223849622
```

- [01~02행] 두 변수 사이의 상관 계수 구하기
 - 01행 survived와 adult_male 변수 사이의 상관 계수를 구함
 - 02행 survived와 fare 변수 사이의 상관 계수를 구함

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

1. 산점도로 상관 분석 시각화하기

```
01 >>> sns.pairplot(titanic, hue = 'survived')
<seaborn.axisgrid.PairGrid object at 0x000001710D852A58>
02 >>> plt.show()
```

- [01~02행] 변수 간의 상관 분석 시각화를 위해 pairplot() 그리기
 - 01행 pairplot() 함수를 사용하여 타이타닉 데이터의 차트를 그림, hue는 종속 변수를 지정
 - 02행 pairplot을 나타냄

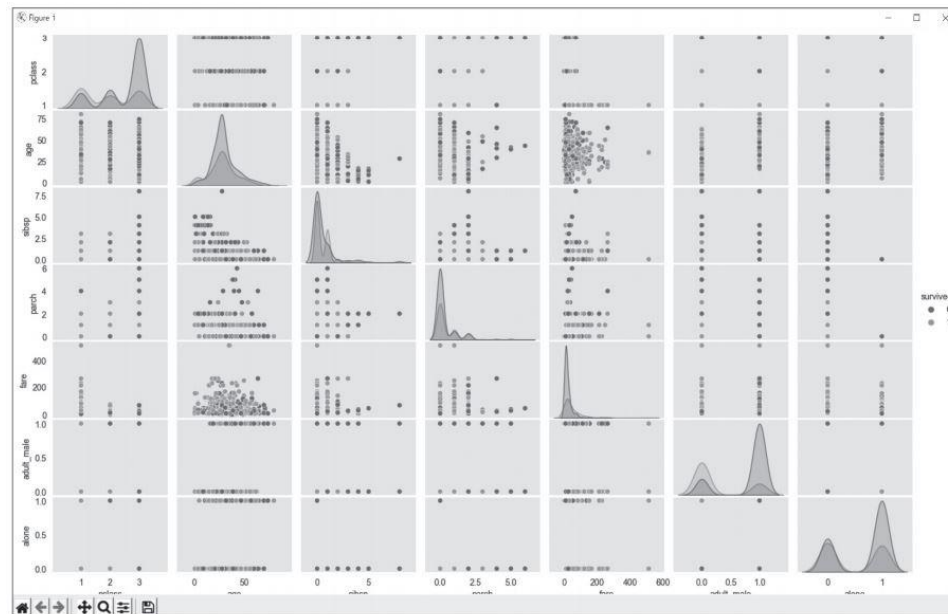


그림 7-12 pairplot() 함수를 이용한 산점도

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

1. 산점도로 상관 분석 시각화하기

```
01 >>> sns.pairplot(titanic, hue = 'survived')
<seaborn.axisgrid.PairGrid object at 0x000001710D852A58>
02 >>> plt.show()
```

- [01~02행] 변수 간의 상관 분석 시각화를 위해 pairplot() 그리기
 - 01행 pairplot() 함수를 사용하여 타이타닉 데이터의 차트를 그림, hue는 종속 변수를 지정
 - 02행 pairplot을 나타냄

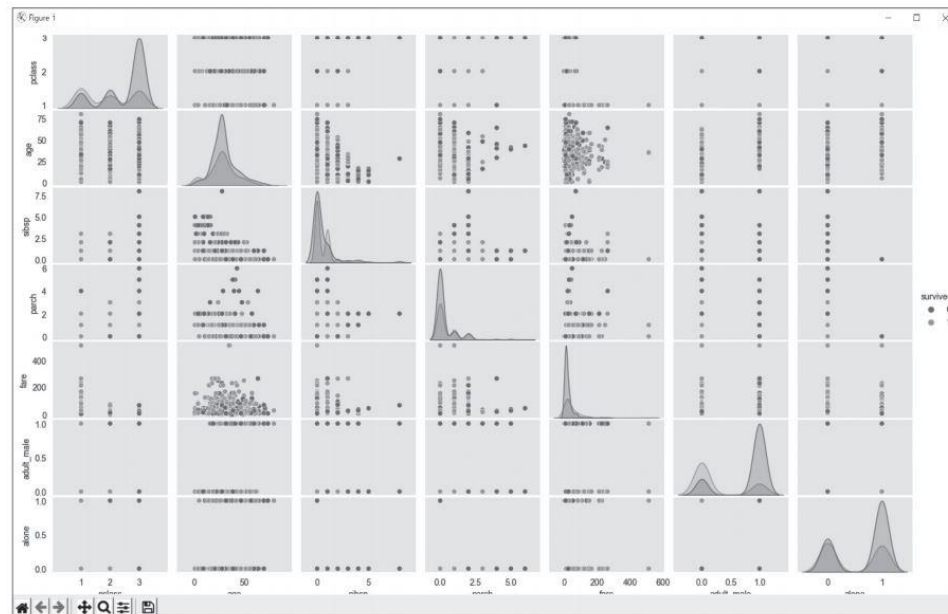


그림 7-12 pairplot() 함수를 이용한 산점도

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

2. 두 변수의 상관관계 시각화하기

```
01 >>> sns.catplot(x = 'pclass', y = 'survived', hue = 'sex', data = titanic, kind = 'point')
      <seaborn.axisgrid.FacetGrid object at 0x000001DD44EB4B88>
02 >>> plt.show()
```

- [01~02행] 생존자의 객실 등급과 성별 관계를 catplot()로 그리기
 - 01행 catplot() 함수를 사용하여 pclass와 survived 변수의 관계를 차트로 그림
hue인자를 이용하여 종속 변수를 sex로 지정
 - 02행 catplot을 나타냄

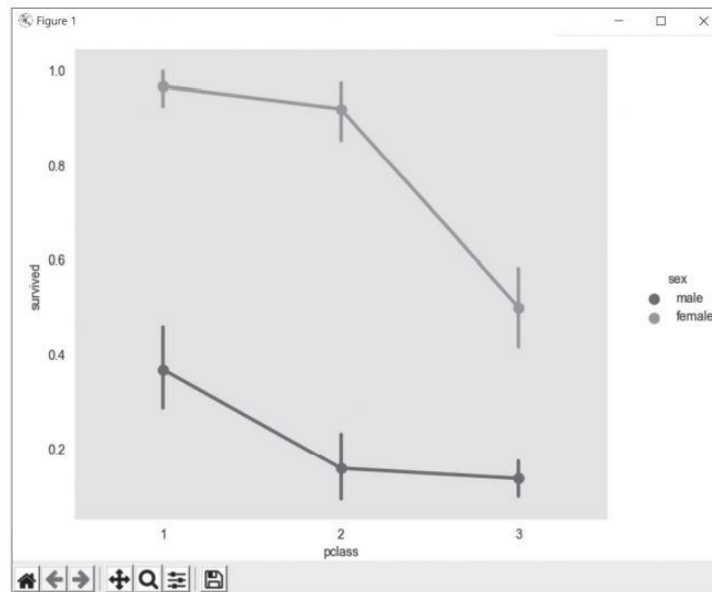


그림 7-13 객실 등급과 생존의 상관관계를 나타내는 catplot 차트

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

2. 변수 사이의 상관 계수를 히트맵으로 시각화하기

```
01 >>> def category_age(x):  
    if x < 10:  
        return 0  
    elif x < 20:  
        return 1  
    elif x < 30:  
        return 2  
    elif x < 40:  
        return 3  
    elif x < 50:  
        return 4  
    elif x < 60:  
        return 5  
    elif x < 70:  
        return 6  
    else:  
        return 7
```

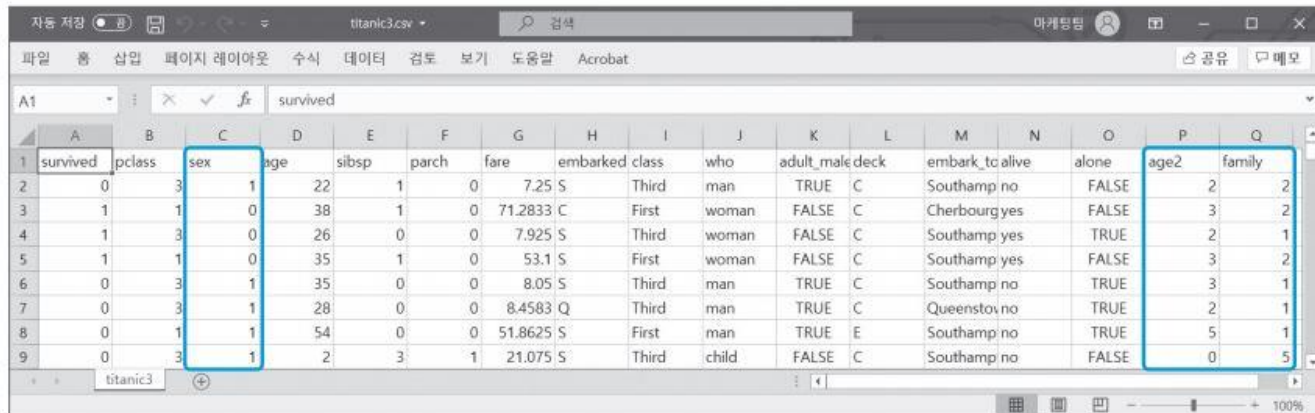
```
02 >>> titanic['age2'] = titanic['age'].apply(category_age)  
03 >>> titanic['sex'] = titanic['sex'].map({'male':1, 'female':0})  
04 >>> titanic['family'] = titanic['sibsp'] + titanic['parch'] + 1  
05 >>> titanic.to_csv('C:/Users/kmj/My_Python/7장_data/titanic3.csv', index =  
    False)  
06 >>> heatmap_data = titanic[['survived', 'sex', 'age2', 'family', 'pclass',  
    'fare']]  
07 >>> colormap = plt.cm.RdBu  
08 >>> sns.heatmap(heatmap_data.astype(float).corr(), linewidths = 0.1, vmax  
    = 1.0, square = True, cmap = colormap, linecolor = 'white', annot = True,  
    annot_kws = {"size": 10})  
    <matplotlib.axes._subplots.AxesSubplot object at 0x000001DD4C8DBF88>  
09 >>> plt.show()
```


02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

3. 변수 사이의 상관 계수를 히트맵으로 시각화하기

- [01~02행] age를 카테고리 값으로 바꾸어 age2 변수로 추가하기
 - 01행 10살 단위로 등급을 나누어 0~7의 값으로 바꿔주는 category_age 함수를 작성
 - 02행 category_age 함수를 적용하여 새로운 age2 열을 만들어 추가
 - 03행 성별을 male/female에서 1/0으로 치환
 - 04행 가족의 수를 구하여 family 열을 추가
 - 05행 수정된 데이터프레임을 titanic3.csv로 저장



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|----------|--------|-----|-----|-------|-------|---------|----------|-------|-------|------------|------|-----------|-------|-------|------|--------|
| 1 | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | embark_to | alive | alone | age2 | family |
| 2 | 0 | 3 | 1 | 22 | 1 | 0 | 7.25 | S | Third | man | TRUE | C | Southamp | no | FALSE | 2 | 2 |
| 3 | 1 | 1 | 0 | 38 | 1 | 0 | 71.2833 | C | First | woman | FALSE | C | Cherbourg | yes | FALSE | 3 | 2 |
| 4 | 1 | 3 | 0 | 26 | 0 | 0 | 7.925 | S | Third | woman | FALSE | C | Southamp | yes | TRUE | 2 | 1 |
| 5 | 1 | 1 | 0 | 35 | 1 | 0 | 53.1 | S | First | woman | FALSE | C | Southamp | yes | FALSE | 3 | 2 |
| 6 | 0 | 3 | 1 | 35 | 0 | 0 | 8.05 | S | Third | man | TRUE | C | Southamp | no | TRUE | 3 | 1 |
| 7 | 0 | 3 | 1 | 28 | 0 | 0 | 8.4583 | Q | Third | man | TRUE | C | Queenstov | no | TRUE | 2 | 1 |
| 8 | 0 | 1 | 1 | 54 | 0 | 0 | 51.8625 | S | First | man | TRUE | E | Southamp | no | TRUE | 5 | 1 |
| 9 | 0 | 3 | 1 | 2 | 3 | 1 | 21.075 | S | Third | child | FALSE | C | Southamp | no | FALSE | 0 | 5 |

그림 7-14 titanic3.csv 파일에서 치환된 내용과 추가된 내용 확인

02. [상관 분석 + 히트맵] 타이타닉호 생존율 분석하기

■ 결과 시각화

3. 변수 사이의 상관 계수를 히트맵으로 시각화하기

- [06~09행] 상관 분석 결과를 히트맵으로 나타내기
 - 06행 히트맵에 사용할 데이터를 추출
 - 07행 히트맵에 사용할 색상맵을 지정
 - 08행 `corr()` 함수로 구한 상관 계수로 히트맵을 생성
 - 09행 생성한 히트맵을 나타냄

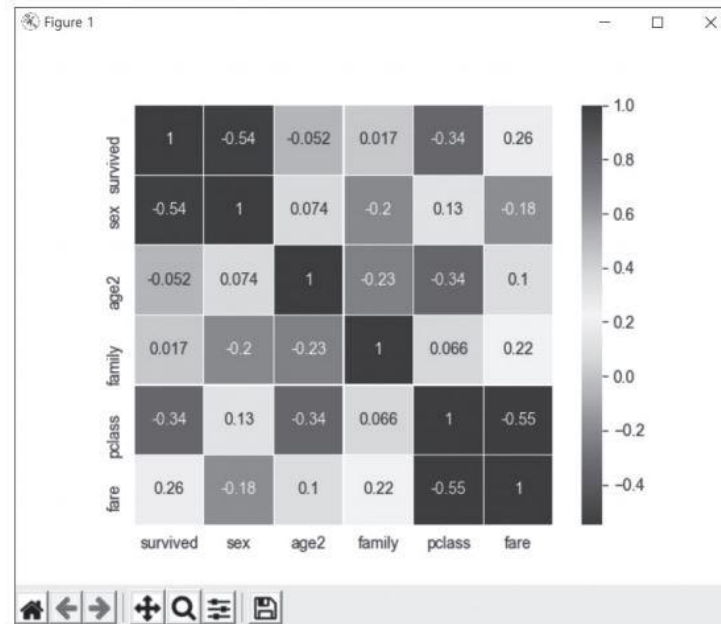


그림 7-15 상관 분석에 대한 히트맵 시각화